INTL-0447-US
(P9445)

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: **ADAPTIVE VIDEO SCALER**

INVENTOR: **PAUL S. GRYSKIEWICZ**

Express Mail No.: EL594060052US

Date: August 31, 2000

ADAPTIVE VIDEO SCALER

Background

This invention relates to video scaling operations.

A digital video scaler is a device that accepts an input image and creates a new output image of different dimensions. A digital video scaler can scale in either the horizontal or the vertical direction.

One objective of digital video scalers is to avoid aliasing artifacts. When the video signal is sampled, the signal is typically sampled at greater than twice the frequency of the signal, known as the Nyquist sampling rate. This ensures the aliasing of the signal is avoided. Aliasing causes unwanted distortion of the original signal. For example, a high-frequency signal, sampled too infrequently, may contain low frequencies that really aren't there.

Whether the size of a video image is increased or decreased, aliasing may be a problem. For example, where an image is scaled down 2:1, say, by discarding half the pixels representing the image, the sampling rate has also been cut in half relative to the original Nyquist sampling rate. The image after scaling may possibly include aliasing artifacts.

Digital video scalers thus typically employ finite impulse response (FIR) filters to scale the video images. An FIR filter receives many data points, or pixels, to generate a single output pixel. A simple averaging FIR filter may receive two or more pixels, average them, and produce an output pixel, for example. The result is a filtered signal.

To scale a signal, the FIR filter receives discrete representations of the signal, or data points, one after the other. The FIR filter performs operations on the data points to produce a resulting data point. Once the signal is filtered, the scaler may discard some of the pixels, yet the resulting scaled output is free of aliasing artifacts.

For video images, the data points are typically pixels. For example, under one digital video standard, an image is sampled at 13.5 MHz. This sampling rate produces 30 frames (or 60 interlaced fields) of image data where the frames include a matrix of pixels, with 720 horizontal pixels by 480 vertical pixels or 480 rows of pixels. Each pixel is received sequentially, starting with all the pixels on the first row, then all the pixels on the second row and so on.

To scale a video image in the horizontal direction, the FIR filter receives each pixel in a selected row, until the FIR filter is filled. The ordering of the pixels received is convenient for horizontal scaling. A simple first-in-first-out (FIFO) memory may receive the pixels into the FIR filter, for example.

Scaling the video image in the vertical direction, however, is not as simple as with horizontal scaling. The FIR filter does not receive the pixels in a convenient way for vertical scaling. Under the above digital video standard, for example, the vertical scaler receives a first pixel in the first row. The next 719 pixels are in the horizontal, not vertical, direction of the frame. Thus, the vertical scaler must wait for the 720th pixel after the first pixel, discard 720 pixels immediately following that then retrieve every 720th pixel until all the vertical pixels are retrieved.

Since the video data is sent horizontally, vertical scalers typically retrieve the pixels from line memories rather than as the original input stream. The vertical scaler may retrieve each vertical pixel from the line memories to perform the scaling operation.

The cost impact of vertical scaling is evident. Higher-quality FIR filters generally include more taps than lower quality FIR filters. Since the amount of available data for processing is less for vertical scaling than for horizontal scaling, vertical scalers typically employ less sophisticated FIR filters (e.g., ones with fewer taps) while horizontal scalers operating on the same video data stream may include many more taps. Where high-quality horizontal scaling may be achieved with a simple FIFO, vertical scaling of a similar quality requires a substantial amount of memory.

Thus, there is a continuing need to improve the efficiency of video scaling operations.

Brief Description of the Drawings

Figure 1 is a block diagram of a system according to one embodiment of the invention;

Figure 2 is a block diagram of the memory according to one embodiment of the invention;

Figure 3 is a block diagram of a finite impulse response filter according to one embodiment of the invention;

Figures 4A-4C are diagrams illustrating storage of the video data stream following horizontal scaling according to a prior art embodiment;

Figures 5A-5C are diagrams illustrating storage of the video data stream following horizontal scaling according to one embodiment of the invention;

Figure 6 is a flow diagram of a horizontal scaling operation according to one embodiment of the invention; and

Figure 7 is a flow diagram of a vertical scaling operation according to one embodiment of the invention.

Detailed Description

In accordance with the several embodiments described herein, an adaptive filter features both horizontal and vertical scaling of an incoming video data stream. The adaptive filter automatically changes the density of data stored in the available memory in real time, following horizontal scaling, such that more efficient vertical scaling may subsequently be performed. Where the amount of stored data increases, the vertical scaler may accordingly support additional taps, allowing for better alias suppression. Where the size of the vertical scaler is increased, the adaptive filter may dynamically configure new coefficients for the vertical scaler. The efficient use of the memory during scaling operations may permit lower-cost alternatives, such as on-chip memory, to be used in the adaptive filter.

In Figure 1, according to one embodiment, an adaptive filter 100 receives a video data stream 20 for performing filtering operations. The adaptive filter 100 may receive the video data stream 20 from a video decoder (not shown) or from an expansion port used to capture the video (not shown).

In one embodiment, the video data stream 20 is a digital video stream. A digital video stream consists of video pixels being transmitted sequentially in the horizontal direction. Pixels are sent from left to right, then after one row is transmitted, the next row is sent until the entire image has been sent. For example, one digital video signal transmits thirty frames per second, where each frame consists of 720 active horizontal video pixels and 480 rows or vertical pixels.

In one embodiment, the adaptive filter 100 receives the video data stream 20 into a first-in-first-out (FIFO) memory 18, before being received into a horizontal scaler 26. In one embodiment, the horizontal scaler 26 is a finite impulse response (FIR) filter, which receives pixels from the FIFO 18 corresponding to the incoming video data stream 20. FIR filters are described in connection with Figure 3, below. In one embodiment, the FIFO 18 stores a number of pixels corresponding to the number of taps (size) of the FIR filter located in the horizontal scaler 26. The horizontal scaler 26 produces a horizontally scaled video data stream 30.

In one embodiment, the horizontally scaled video data stream 30 is stored in a memory 22 of the adaptive filter 100. The memory 22 may be a random access memory such as dynamic random access memory (DRAM), synchronous DRAM (SDRAM), static ram (SRAM), or Rambus® DRAM (RDRAM). The memory 22 comprises a plurality of bits for storing data. In one embodiment, the memory 22 is embedded in other circuitry of the system 100, as on-chip memory. On-chip memory may be preferred over a dedicated memory for designs where board space and cost are considerations, in some embodiments.

The memory 22 may be organized to facilitate access thereto, in some embodiments. For example, as shown in Figure 2, the memory 22 is divided into a plurality of line memories 48, wherein each line memory 48 is a predetermined length. In one embodiment, the line memory 48 is an amount of memory that may store a single row of video data.

Accordingly, the size of the line memory 48 may change upon receiving each new video data stream into the system 100. For example, for a video signal with a frame size of 720 x 480 comprising three bytes of data per pixel, the line memory 48 is 2160 bytes long (720 x 3). A video signal with 800 x 600 resolution frames which includes two bytes of data per pixel stores each row of data in a 1600-byte line memory 48 (800 x 2). The line memory 48 is dynamically adjusted to accept a row worth of video data, according to one embodiment.

A memory controller 24 is coupled to the memory 22, for controlling memory access. In one embodiment, the memory controller 24 may be programmed to change how rows of pixel data are both stored and retrieved in the line memories 48. In one embodiment, the memory controller 24 is programmed, based upon the scaling operation performed in the horizontal scaler 26, to facilitate efficient vertical scaling operations. Accordingly, a plurality of registers 32 are coupled to the memory controller 24. One register 32, for example, may identify the number of rows of pixel data stored in each line memory 48. A second register 32 may indicate offset addresses in the line memory 48 for accessing each row. These examples illustrate two of a number of uses for the registers 32 by the memory controller 24, to facilitate memory access before and after scaling operations.

In one embodiment, the memory 22 is coupled to a vertical scaler 36. Like the horizontal scaler 26, the vertical scaler 36 uses an FIR filter which scales frames of the incoming video data stream 20 in the vertical direction, producing a scaled video data stream 40. In one embodiment, the scaled video data stream 40 is scaled in both the horizontal and vertical directions.

A scaling control unit 28 is coupled to both the horizontal scaler 26 and the vertical scaler 36, for dynamically controlling the scaling operations in real time. In one embodiment, the scaling control unit 28 is a processor-based system which controls the filter size (number of taps) for each of the horizontal and vertical scalers 26 and 36. The scaling control unit 28 further includes storage 16, such as for software 200 running in the scaling control unit 28.

In one embodiment, the scaling control unit 28 includes a look-up table (LUT) 38, for retrieving the tap coefficients for the FIR filters. In addition to controlling the horizontal scaler 26 and the vertical scaler 36, the scaling control unit is coupled to the memory controller 24, for automatically changing the number of rows of pixel data that are stored in the memory 22.

In one embodiment, the adaptive filter 100 performs video scaling operations on the incoming video data stream 20. The adaptive filter 100 may thus be used where a video image is "down-sized," such as for supporting picture-in-picture (PIP), or to add additional graphics or other video data to the video image, such as display of stock quotes simultaneous with a television broadcast, for example.

Instead of processing the video data stream 20 through an FIR filter, a less sophisticated approach to scaling may be to simply discard every other pixel of the incoming video data stream 20, or, in other words, perform no filtering at all. As explained above, however, the resulting image may include aliasing artifacts which obscure the image.

FIR filters avoid aliasing artifacts by using as many of the data samples as possible to perform scaling operations. An FIR filter may, for example, receive data samples, in this case, pixels, from the video data stream 20 and multiply the data samples by constant values, known as tap coefficients. The products of the multiplication are then summed together to produce a result. (The FIR filter essentially performs a "dot product," a well-known mathematical operation.) Because the high-frequency components are filtered out (reduced), the possibility of aliasing artifacts in the scaled image may be reduced.

In Figure 3, an FIR filter 70 according to one embodiment receives an incoming signal 50, extracts a plurality of data samples 52 from the incoming signal, and multiplies each data sample 52 by a coefficient 62. The FIR filter 70 includes N data samples 52 as well as N coefficients 62, each producing a product 54. The products 54 are then summed together to produce a filtered output signal 60.

The coefficients 62 of the FIR filter 70 are known as "taps." The size of the FIR filter 70 corresponds to the number of taps. Thus, the FIR filter 70 of Figure 3 is an N-tap filter.

In designing the FIR filter 70, a desired output signal, or frequency response, is determined and, based upon the desired response, the coefficients 62 are calculated. The filtered output signal 60 produced using the coefficients 62 is typically an approximation of the desired frequency response. The coefficients 62 change depending on the size of N. For example, a two-tap FIR filter 70 employs different coefficients 62 than a three-tap FIR filter 70. Having more taps tends to produce a higher quality filtered output signal 60, up to a point. Increasing the number of taps also increases the computational operations performed by the FIR filter 70. Finding an optimal number of taps for the operation being performed is thus a key design issue.

Because a frame of video data is collected and transmitted, row by row, an FIR filter performing horizontal scaling has ready access to the data samples used to compute the filtered output signal. Such an FIR filter may typically include 65 to 80 taps. An FIR filter performing vertical scaling, by contrast, typically includes only two to five taps, due to the additional delay associated with retrieving the data samples.

According to one embodiment, the adaptive filter 100 may optimize the use of the memory 22 following a horizontal scaling operation by varying the density of stored information. By effectively increasing the available memory 22 accessible to the vertical scaler 36, the number of data samples 52 that may be stored in the memory 22 is increased. Accordingly, the size of the FIR filter 70 in the vertical scaler 36 may be increased for performing vertical scaling. By increasing the size (e.g., the number of taps) of the vertical scaler 36 without increasing the amount of memory 22 needed to perform vertical scaling, the effective cost of the adaptive filter 100 is decreased while the quality of the filtered output signal 60 may actually improve.

In some prior art systems, an entire frame of video data may be stored in a frame buffer memory prior to performing any scaling or following the horizontal scaling operation. This results in at least one video frame of delay, which is not a real-time operation. Further, memory for storing an entire frame of video data is typically not small. For example, a video frame with a resolution of 720 x 480 which stores three bytes per pixel is over one megabyte in size. On-board memories, however, typically store up to 2 kilobytes, 4 kilobytes, or 8 kilobytes of data.

In contrast, according to one embodiment, the adaptive filter 100 performs real-time operations during horizontal and vertical scaling. The operation to increase the density of stored data in the memory 22, following the horizontal scaling operation, is performed in real-time. Likewise, any adjustment to the size of the FIR filter 70 in the vertical scaler 36 is made in real-time. Additionally, the adaptive filter 100 makes efficient use of the memory 22, making it possible for smaller memories, such as on-board memory, to be employed for performing these real-time operations.

In Figures 4A-4C, pixels from the incoming video data stream 20 are stored in the memory 22 according to a typical prior art embodiment. Where no horizontal scaling is performed, e.g., 1:1 horizontal scaling, a first row 58 of the video data stream is stored in the first line memory 48, followed by a second row 58 stored in the second line memory 48, and so on, one row after another. For example, under the digital video standard described above, 720 pixels, corresponding to the first row 58 of the incoming video data stream 20, are stored in the first line memory 48 of the memory 22. Thus, there is direct correspondence between the line memory 58 number (e.g., first, second, third...) and the row 48 of the frame being stored.

Where the horizontal scaler 26 performs 2:1 horizontal scaling, the resulting pixels stored in the memory 22 occupy only half of the available space, as illustrated in Figure 4B. The first row 58 of video data is stored in the first line memory 48, as before. Because the row 58 occupies only half of the available line memory 48, a second row 58 may be stored in the first line memory 48. This is not what happens, however. Instead, the second row 58 is stored in the second line memory 48, not the first. As with 1:1 horizontal scaling, the first row 58 is stored in the first line memory 48, the second row 58 is stored in the second line memory 48, the third row 58 is stored in the third line memory 48, and so on. Where 2:1 horizontal scaling is performed, half the available memory 22 is unused. Where 4:1 horizontal scaling is performed, only one-fourth of the memory 22 stores the pixel data, leaving three-fourths of the memory 22 empty, as depicted in Figure 4C.

According to one embodiment, the adaptive filter 100 optimizes the use of the memory 22 by increasing the density of data stored following horizontal scaling operations. For example, in Figure 5A, following 2:1 horizontal scaling, the first and second rows 58 are stored in the first line memory 48; the third and fourth rows 58 are stored in the second line memory 48; the fifth and sixth rows 58 are stored in the third line memory 48, and so on. Accordingly, twice as many rows 58 of pixel data are stored in each line memory 48 as during the 2:1 horizontal scaling operation described in Figure 4B.

In Figure 5B, following a 3:1 scaling operation, the first, second and third rows 58 are stored in the first line memory 48, the fourth, fifth and sixth rows 58 are stored in the second line memory, and so on. Three times as many rows 58 are stored, in one embodiment, as when no scaling operation takes place. Likewise, in Figure 5C, four times as many rows 58 are stored in each line memory 48 following a 4:1 scaling operation.

In one embodiment, the adaptive filter 100 performs vertical scaling following the horizontal scaling operation. By first storing the pixels of the horizontally scaled video data stream 30 in the manner shown in Figures 5A-5C, the number of available data samples 52 to be used by the FIR filter 70 of the vertical scaler 36 is increased. For example, in Figures 4A-4C, the same number of rows 58 is stored in the memory 22, whether horizontal scaling was performed or not. In each case, four rows 58 of pixel data, and thus four data samples 52, are available for vertical scaling. In Figure 5A, however, because eight lines 58 are stored in the memory 22, eight data samples 52 are available for vertical scaling. Likewise, in Figures 5B and 5C, twelve and sixteen data samples 52, respectively, are available for scaling in the vertical direction.

The memory controller 24 controls accesses to and from the memory 22. In one embodiment, the scaling control unit 28 directs the memory controller 24 to change where each row 58 is stored, according to a prior horizontal scaling operation. Put another way, the number of rows 58 stored for each line memory 48 may be adjusted by the scaling control unit 28. In one embodiment, the registers 32 are updated by the scaling control unit 28 following horizontal scaling. The memory controller 24 may access the registers 32 to determine how subsequent accesses to the memory 22 are made.

Further, the scaling control block 28, in one embodiment, controls both the number of taps and the coefficients 62 used in the FIR filter 70 of the vertical scaler 36. Once the number of rows 58 stored in the memory 22 is known, the number of taps used in the FIR filter 70 of the vertical scaler 36 may be increased accordingly.

In one embodiment, the scaling control unit 28 recalculates the coefficients 62 of the FIR filter 70 in accordance with the increased filter size. In one embodiment, the scaling control unit 28 supplies the new coefficients 62 to the vertical scaler 36 by retrieving the coefficients 62 from the LUT 38. In a second embodiment, the scaling control unit 28 retrieves the coefficients 62 from a source external to the adaptive filter 100.

The registers 32 of the adaptive filter 100 may further supply the scaling control unit 28 as well as the memory controller 24 with information relating to the scaling operations being performed. For example, the registers 32 may indicate the size of the scaling operation being performed (e.g., 3:1), the size of the memory 22, the current number of taps in the vertical scaler 36, and so on.

In Figure 6, a flow diagram illustrates operation of the software program 200, running in the scaling control unit 28, to direct the horizontal scaler 26 and the memory controller 24 to store rows 58 of video data such as depicted in Figures 5A-5C, above. The adaptive filter 100 receives the video data stream 20, such as into the FIFO 18 of Figure 1 (block 202). The horizontal scaler 26 performs an X:Y scaling operation on the video data stream 20 (block 204), where Y/X represents the scaling ratio. For example, a 4:1 scaling operation would scale down the data by a factor of four, for a ¼ scaling ratio.

Once the scaling ratio is determined, the scaling control unit 28 uses the ratio to direct the horizontal scaler 26 to store the scaled data 30 in the memory 22 accordingly (block 206). In one embodiment, X/Y rows of the scaled video data 30 are stored in each line memory (block 210), until all line memories 48 have been accessed. So, for 4:1 scaling, each line memory 48 stores four rows, where previously only a single row was stored. In this manner, the horizontally scaled video data 30 is optimally stored in the memory 22.

The scaling ratio does not always produce an integer number of rows in the line memory 48. Thus, in some embodiments, a portion of a row may be stored in one line memory 48 following horizontal scaling. For example, where 10:4 horizontal scaling is performed, a 2/5 scaling ratio results, such that 2 ½ rows of the scaled video data 30 are stored in each line memory 48. In other embodiments, the scaling ratio is such that different line memories 48 may store different number of rows.

In some embodiments, the vertical scaler 36 may access additional data samples as a result of the optimal storage. In one embodiment, the software 200 of the scaling control unit 28 may direct the vertical scaler 36, as depicted in the flow diagram of Figure 7. First, the vertical scaler 36 determines the scaling ratio X:Y for a prior horizontal scaling operation that was performed (block 252). In one embodiment, the vertical scaler 36 retrieves this information from the scaling control unit 28. As during the prior store operation (Figure 6), the vertical scaler 36 accesses the line memories 48 of the memory 22, starting from the first line memory 48.

In one embodiment, from each line memory 48, X/Y data samples are retrieved (block 254), e.g., the inverse of the scaling ratio. Depending on the value of X/Y, the number of data samples may be different for different line memories 48 of the memory 22. For example, where a 10:2 scaling operation is performed, two and ½ rows of data are stored in each line memory 48, according to one embodiment. The ½-row may or not include the desired data sample, however. Thus, a line memory 48 may provide three data samples, while a subsequent line memory 48 provides two data samples, then the following line memory provides three data samples, and so on.

Once the last line memory 48 is reached, the data samples are sent to the FIR filter of the vertical scaler 36 (block 256). Thus, according to one embodiment, the vertical scaler 36 may produce the scaled video data stream 40.

Because the vertical scaler 36 has access to more data samples 52 stored in the memory 22 than would typically be available, the number of taps used by the FIR filter 70 in the vertical scaler 36 may likewise be increased. The number of taps essentially corresponds to the number of coefficients 62 used to generate a product with the data samples 52 prior to performing a summation operation (see Figure 3).

Accordingly, in one embodiment, following the horizontal scaling operation, the scaling control unit 28 supplies the vertical scaler 36 with the number of taps to be used by its FIR filter 70. Likewise, the scaling control unit 28 sends the vertical scaler 36 a new set of coefficients 62, corresponding to the number of taps. In one embodiment, the number of taps used by the FIR filter 70 in the vertical scaler 36 is equal to the number of rows 58 stored in the memory 22 plus one. The number of rows 58 corresponds to the number of available vertical data samples 52, and the additional value accounts for the fact that the current sample retrieved from the video data stream 20 need not be stored before vertical scaling.

The number of available line memories 48 which are accessible to the vertical scaler 36 depend on a number factors, including the speed at which the video data stream 20 is being transmitted, the size of the memory 22, the desired speed of the scaled video data stream 40 and other factors. In some embodiments, the adaptive filter 100 may flexibly be implemented in a variety of environments.

The scaling ratio for both horizontal scaling and vertical scaling are considered by the adaptive filter 100, according to several embodiments. For example, in one embodiment, the adaptive filter 100 performs the operations described herein when the scaling ratio of the horizontal scaler 26 is identical to the scaling ratio of the vertical scaler 36. In another embodiment, the adaptive filter operates where the horizontal scaling ratio is not the same as the vertical scaling ratio.

Thus, in one embodiment, an adaptive filter may optimize the storage of a scaled data stream such that more robust scaling may be performed. Where a scaling operation has been performed on a data stream, storage of the scaled data may be increased in some embodiments. Where the number of available data samples for subsequent scaling operations has increased, the size of FIR or other filters used during the subsequent scaling may be increased. Coefficients may likewise be supplied to the filter where its size is increased. In some embodiments, subsequent scaling operation enhances the scaled output without adding cost to the adaptive filter.

While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of the present invention.